Задача A. Альфа, бета, гамма…

Имя входного файла:	`abg.in`
Имя выходного файла:	`abg.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Выведите английское название n-ой буквы греческого алфавита.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 24$).

Формат выходного файла

В выходной файл выведите английское название соответствующей буквы греческого алфавита строчными буквами.

Примеры

abg.in	abg.out
1	alpha
24	omega

Примечание

Необходимую информацию вы можете найти на странице `http://en.wikipedia.org/wiki/Greek_alphabet#Letter_names`.

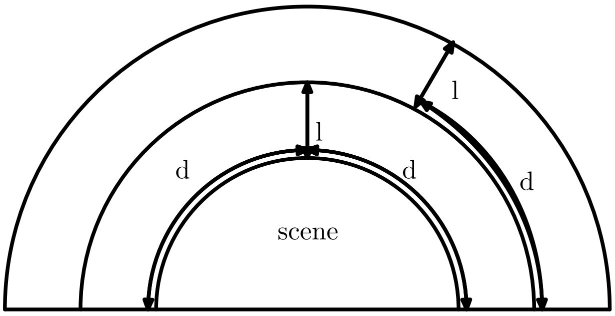
Задача B. Амфитеатр

Имя входного файла:	`amfi.in`
Имя выходного файла:	`amfi.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Как известно, древние греки очень любили строить амфитеатры. Амфитеатр — это здание в форме полукруга, часть которого, а именно полукруг с тем же центром, но с меньшим радиусом, и лежащий внутри амфитеатра, занята сценой. Остальное же место отводится под сидения.

Недавно греки современные решили построить новейший амфитеатр с радиусом R метров и радиусом сцены r метров. Тепеть они хотят узнать, сколько же зрителей будет вмещать сооружение.

Зрители сидят на скамьях толщины l сантиметров вдоль окружности лицом к сцене. Каждая скамья полностью находится внутри амфитеатра. Один грек, садясь на скамью занимает d сантиметров (вдоль внутренней кромки скамьи). При этом греки могут сидеть вплотную.



Формат входного файла

Четыре натуральных числа R, r, l и d, не превосходящие 1000.

Формат выходного файла

В выходной файл выведите одно число — максимальное количество греков, которые могут уместиться в амфитеатре.

Примеры

amfi.in	amfi.out
2 1 100 314	1
100 10 200 100	7611

Задача C. k-ая порядковая статистика. Обратная задача

Имя входного файла:	`invkth.in`
Имя выходного файла:	`invkth.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Пусть задано n попарно различных целых чисел $a_1, a_2, \ldots, a_n$. Назовем k-ой порядковой статистикой среди них такое число $x = a_j$ (для некоторого j), что среди $a_1, \ldots, a_n$ существует ровно $k-1$ число, меньшее x.

Задача нахождения k-ой порядковой статистики достаточно хорошо изучена, для ее решения существует множество алгоритмов. Поэтому ваша задача состоит в другом. Заданы числа n, k и x. Необходимо построить набор из n попарно различных целых чисел, в котором x будет являться k-ой порядковой статистикой.

Формат входного файла

Первая строка входного файла содержит три целых числа: n, k, x ($1 \le k \le n \le 10^5$, $|x| \le 10^9$).

Формат выходного файла

В первой строке выходного файла выведите искомые числа a_i. Соседние числа отделяйте друг от друга пробелами. Все a_i должны быть попарно различными и не должны превосходить 10^9 по абсолютной величине. Гарантируется, что набор a_i, удовлетворяющий описанным требованиям, существует.

Примеры

invkth.in	invkth.out
4 2 3	1 3 4 5

Задача D. Прямые

Имя входного файла:	`lines.in`
Имя выходного файла:	`lines.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На плоскости задано n различных прямых, каждая из которых либо вертикальна, либо горизонтальна. Необходимо найти число их точек пересечения.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 10^5$). Каждая из последующих n строк описывает одну прямую и содержит четыре целых числа, не превосходящих 10^9 по абсолютной величине: x_1, y_1, x_2, y_2 — координаты двух различных точек, лежащих на этой прямой.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

lines.in	lines.out
1 0 0 1 0	0
3 0 0 1 0 0 1 1 1 0 0 0 1	2

Задача E. Кораблевождение

Имя входного файла:	`navi.in`
Имя выходного файла:	`navi.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Кораблевождение — наука, излагающая правила о том, как вести корабль. Кораблевождение состоит из нескольких разделов, изучающих прокладку маршрута, навигацию, непосредственно управление кораблем.

Вообще, движение корабля в море характеризуется достаточно большим набором величин, но в этой задаче будет рассматриваться упрощенная модель. Направления движения корабля будет характеризоваться одним целым числом — его *курсом*, который измеряется в пределах от 0 (включительно) до 360 (не включительно) градусов. Курс отсчитывается к востоку от направления на север.

В процессе движения корабль может осуществлять повороты. Поворот осуществляется на некоторое число градусов на правый или на левый борт. Ваша задача — написать программу, вычисляющую новый курс корабля после поворота.

Формат входного файла

Первая строка входного файла содержит два целых числа x ($0 \le x < 360$) и y ($|y| \le 180$) — соответственно, исходный курс корабля и угол, на который был совершен поворот. Если $y > 0$, то поворот совершается на правый борт, иначе — на левый.

Формат выходного файла

В выходной файл выведите курс корабля после поворота.

Примеры

navi.in	navi.out
359 10	9
359 -10	349

Задача F. Числа

Имя входного файла:	`numbers.in`
Имя выходного файла:	`numbers.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Задано n положительных целых чисел a_1, a_2, $\ldots$, a_n. Необходимо выбрать из них как можно больше чисел так, чтобы сумма выбранных чисел делилась на 3.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 10^5$). Вторая строка входного файла содержит n целых чисел a_1, $\ldots$, a_n, находящихся в диапазоне от 0 до 10^9.

Формат выходного файла

В первой строке выходного файла выведите количество выбранных чисел. Во второй строке выведите их номера во входном файле в порядке возрастания. Числа нумеруются в порядке их перечисления во входном файле натуральными числами от 1 до n.

Если ответов несколько, выведите любой.

Примеры

numbers.in	numbers.out
3 4 5 6	3 1 2 3
3 4 2 5	2 1 2

Задача G. Последовательность

Имя входного файла:	`seq.in`
Имя выходного файла:	`seq.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Пусть x — натуральное число. Обозначим как $s(x)$ сумму цифр квадратов его делителей. Например, $s(6) = 1 + 4 + 9 + (3 + 6) = 23$, $s(10) = 1 + 4 + (2 + 5) + (1 + 0 + 0) = 13$.

Рассмотрим теперь последовательность $a_1 = x$, $a_2 = s(x)$, $a_3 = s(s(x))$, $\ldots$, $a_n = s(a_{n-1})$, $\ldots$.

Скажем, что последовательность a_n имеет длину периода L и длину предпериода P, если при любом $i > P$ верно равенство $a_i = a_{i+L}$, при этом и L и P минимальные числа, удовлетворяющие этому условию.

Ваша задача — написать программу, вычисляющую длину периода и предпериода описанной выше последовательности $a_1 = x$, $a_2 = s(x)$, $a_3 = s(s(x))$, $\ldots$, $a_n = s(a_{n-1})$, $\ldots$

Формат входного файла

Входной файл содержит целое число x ($1 \le x \le 10^9$).

Формат выходного файла

В выходной файл выведите два числа: длину предпериода P и длину периода L. Если хотя бы одно из чисел P или L превышает 10^3, то выведите в выходной файл `-1 -1`.

Примеры

seq.in	seq.out
16	3 7
1	0 1

Задача H. Сумма на отрезке

Имя входного файла:	`sum.in`
Имя выходного файла:	`sum.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Задано n целых чисел a_1, a_2, $\ldots$, a_n. Суммой на отрезке $[l, r]$ $(1 \le l \le r \le n)$ назовем число, равное $\sum\limits_{i=l}^{r} a_i = a_l + \ldots + a_r$.

Напишите программу, которая определяет, существует ли отрезок, сумма на котором равна x.

Формат входного файла

Первая строка входного файла содержит целое число n $(1 \le n \le 1000)$. Вторая строка входного файла содержит n целых чисел a_i, отделенных друг от друга пробелами. Все a_i не превосходят 10^9 по абсолютной величине. Третья строка входного файла содержит целое число x $(|x| \le 10^9)$.

Формат выходного файла

В первой строке выходного файла выведите два числа l и r $(l \le r)$, задающих искомый отрезок. Если сумма ни на одном отрезке не равна x, то выведите `-1 -1`.

Пример

sum.in	sum.out
4 1 4 2 3 5	3 4
4 1 2 3 4 -5	-1 -1